CLAIMS

We claim:

1. A method comprising:

performing a significance propagation pass on a group of coefficients;

creating a data structure that indicates locations of coefficients in the group of coefficients that are to be processed in subsequent passes; and

performing a refinement sub-bitplane pass by

accessing the data structure to obtain information to identify coefficients to be skipped for refinement sub-bitplane pass processing,

accessing a memory storing the group of coefficient using the information to only access coefficients identified as being in the refinement pass, and

coding refinement bits accessed from the memory.

2. The method defined in Claim 1 wherein a plurality of refinement bits are accessed and non-refinement bits are ignored.

3. The method defined in Claim 1 wherein the group of coefficients comprises a code-block.

4. The method defined in Claim 1 wherein creating at least one data structure comprises creating a data structure to describe locations of coefficients in a refinement sub-bitplane pass for the group of coefficients.

5. The method defined in Claim 4 wherein a first data structure is created for refinement bits and a second data structure is created for cleanup bits during the significance propagation pass.

6. The method defined in Claim 4 wherein the data structure includes an indication of each run of refinement bits and the number of coefficients to skip in the group of coefficients before the next run of refinement bits.

7. The method defined in Claim 6 wherein at least one of the indication of each run of refinement bits and the number of coefficients to skip is represented with a variable length code as follows:

count	codeword
1	000
2	001
3	010
4	011
5	1000000000000
6	1000000000001
...	...
4096	1111111111011

8. The method defined in Claim 6 wherein at least one of the indication of each run of refinement bits and the number of coefficients to skip is represented with a variable length code as follows:

count	codeword
1	0
2	01
3	1100
4	1101_0
5	1101_1
6	1110_0000
7	1110_0001
...	...
21	1110_1111
22	1111_0000_0000_0000
23	1111_0000_0000_0001
...	...
4096	1111_1111_1110_1010

9. The method defined in Claim 6 wherein at least one of the indication of each run of refinement bits and the number of coefficients to skip is represented with a gamma code as follows:

count	codeword (γ^1 format)	codeword (γ format)
1	0	0
2	10_0	100
3	10_1	110
4	110_00	10100
5	110_01	10110
6	110_10	11100
7	110_11	11110
8	1110_000	1010100
9	1110_001	1010110
...	...	...
15	1110_111	1111110
16	11110_0000	101010100
32	111110_00000	10101010100
64	111110_000000	1010101010100
128	11111110_0000000	101010101010100
256	111111110_00000000	10101010101010100
512	1111111110_000000000	1010101010101010100
1024	11111111110_000000000	101010101010101010100
2048	111111111110_0000000000	10101010101010101010100
4096	1111111111110_00000000000	101010101010101010101010100

10. The method defined in Claim 1 wherein the indication of each of run of refinement bits is stored as an integer.

11. The method defined in Claim 10 wherein the integer comprises a minimal size integer.

12. The method defined in Claim 1 wherein the indication of a number of coefficients to skip is stored as an integer.

13. The method defined in Claim 10 wherein the integer comprises a minimal size integer.

14. The method defined in Claim 1 wherein the indication of each of run of refinement bits and the number of coefficients to skip are stored as integers.

15. The method defined in Claim 14 wherein the integers comprise minimal size integers.

16. An apparatus comprising:

means for performing a significance propagation pass on a group of coefficients;

means for creating a data structure that indicates locations of coefficients in the group of coefficients that are to be processed in subsequent passes; and

means for performing a refinement sub-bitplane pass by

accessing the data structure to obtain information to identify coefficients to be skipped for refinement sub-bitplane pass processing,

accessing a memory storing the group of coefficient using the information to only access coefficients identified as being in the refinement pass, and

coding refinement bits accessed from the memory.

17. The apparatus defined in Claim 16 wherein a plurality of refinement bits are accessed and non-refinement bits are ignored.

18. The apparatus defined in Claim 16 wherein the group of coefficients comprises a code-block.

19. The apparatus defined in Claim 16 wherein the means for creating at least one data structure comprises means for creating a data

structure to describe locations of coefficients in a refinement sub-bitplane pass for the group of coefficients.

20. The apparatus defined in Claim 19 wherein a first data structure is created for refinement bits and a second data structure is created for cleanup bits during the significance propagation pass.

21. The apparatus defined in Claim 19 wherein the data structure includes an indication of each run of refinement bits and the number of coefficients to skip in the group of coefficients before the next run of refinement bits.

22. The apparatus defined in Claim 21 wherein at least one of the indication of each run of refinement bits and the number of coefficients to skip is represented with a variable length code as follows:

count	codeword
1	000
2	001
3	010
4	011
5	1000000000000
6	1000000000001
...	...
4096	1111111111011

23. The apparatus defined in Claim 21 wherein at least one of the indication of each run of refinement bits and the number of coefficients to skip is represented with a variable length code as follows:

count	codeword
1	0
2	01
3	1100
4	1101_0
5	1101_1
6	1110_0000
7	1110_0001
...	...
21	1110_1111
22	1111_0000_0000_0000
23	1111_0000_0000_0001
...	...
4096	1111_1111_1110_1010

24. The apparatus defined in Claim 21 wherein at least one of the indication of each run of refinement bits and the number of coefficients to skip is represented with a gamma code as follows:

count	codeword (γ^1 format)	codeword (γ format)
1	0	0
2	10_0	100
3	10_1	110
4	110_00	10100
5	110_01	10110
6	110_10	11100
7	110_11	11110
8	1110_000	1010100
9	1110_001	1010110
...	...	...
15	1110_111	1111110
16	11110_0000	101010100
32	111110_00000	10101010100
64	111110_000000	1010101010100
128	11111110_0000000	101010101010100
256	111111110_00000000	10101010101010100
512	1111111110_000000000	1010101010101010100
1024	11111111110_000000000	10101010101010101010100
2048	111111111110_0000000000	101010101010101010101010100
4096	1111111111110_00000000000	10101010101010101010101010100

25. An article of manufacture having one or more recordable media with executable instructions stored thereon which, when executed by a system, cause the system to:

perform a significance propagation pass on a group of coefficients;

create a data structure that indicates locations of coefficients in the group of coefficients that are to be processed in subsequent passes; and

perform a refinement sub-bitplane pass by

accessing the data structure to obtain information to identify coefficients to be skipped for refinement sub-bitplane pass processing,

accessing a memory storing the group of coefficient using the information to only access coefficients identified as being in the refinement pass, and

coding refinement bits accessed from the memory.

26. A method comprising:

performing a significance propagation pass on a group of coefficients;

creating a data structure that indicates locations of coefficients in the group of coefficients that are to be processed in subsequent passes; and

performing a cleanup sub-bitplane pass by

accessing the data structure to obtain information to identify coefficients to be skipped for cleanup sub-bitplane pass processing,

accessing a memory storing the group of coefficient using the information to only access coefficients identified as being in the cleanup pass, and

coding cleanup bits accessed from the memory.

27. The method defined in Claim 26 wherein a plurality of cleanup bits are accessed and non-cleanup bits are ignored.

28. The method defined in Claim 26 wherein the group of coefficients comprises a code-block.

29. The method defined in Claim 26 wherein creating at least one data structure comprises creating a data structure to describe locations of coefficients in a cleanup sub-bitplane pass for the group of coefficients.

30. The method defined in Claim 29 wherein the data structure includes an indication of each run of cleanup bits and the number of coefficients to skip in the group of coefficients before the next run of cleanup bits.

31. The method defined in Claim 30 wherein at least one of the indication of each run of cleanup bits and the number of coefficients to skip is represented with a variable length code as follows:

count	codeword
1	000
2	001
3	010
4	011
5	1000000000000
6	1000000000001
...	...
4096	1111111111011

32. The method defined in Claim 30 wherein at least one of the indication of each run of cleanup bits and the number of coefficients to skip is represented with a variable length code as follows:

count	codeword
1	0
2	01
3	1100
4	1101_0
5	1101_1
6	1110_0000
7	1110_0001
...	...
21	1110_1111
22	1111_0000_0000_0000
23	1111_0000_0000_0001
...	...
4096	1111_1111_1110_1010

33. The method defined in Claim 30 wherein at least one of the indication of each run of cleanup bits and the number of coefficients to skip is represented with a gamma code as follows:

count	codeword (γ' format)	codeword (γ format)
1	0	0
2	10_0	100
3	10_1	110
4	110_00	10100
5	110_01	10110
6	110_10	11100
7	110_11	11110
8	1110_000	1010100
9	1110_001	1010110
...	...	...
15	1110_111	1111110
16	11110_0000	101010100
32	111110_00000	10101010100
64	111110_000000	1010101010100
128	11111110_0000000	101010101010100
256	111111110_00000000	10101010101010100
512	1111111110_000000000	1010101010101010100
1024	11111111110_000000000	10101010101010101010100
2048	111111111110_0000000000	101010101010101010101010100
4096	1111111111110_00000000000	10101010101010101010101010100

34. The method defined in Claim 30 wherein the indication of each of run of cleanup bits is represented with a variable length code.

35. The method defined in Claim 30 wherein the indication of a number of coefficients to skip is represented with a variable length code.

36. The method defined in Claim 30 wherein the indication of each of run of cleanup bits and the number of coefficients to skip are represented with variable length codes.

37. The method defined in Claim 30 wherein the indication of each of run of cleanup bits is stored as an integer.

38. The method defined in Claim 37 wherein the integer comprises a minimal size integer.

39. The method defined in Claim 30 wherein the indication of a number of coefficients to skip is stored as an integer.

40. The method defined in Claim 37 wherein the integer comprises a minimal size integer.

41. The method defined in Claim 30 wherein the indication of each of run of cleanup bits and the number of coefficients to skip are stored as integers.

42. The method defined in Claim 37 wherein the integers comprise minimal size integers.

43. The method defined in Claim 26 wherein creating at least one data structure comprises creating a first data structure to describe locations of coefficients in a refinement sub-bitplane pass for the group of coefficients and a second data structure to describe locations of coefficients in a cleanup sub-bitplane pass for the group of coefficients.

44. The method defined in Claim 43 wherein the group of coefficients comprises a code-block.

45. An apparatus comprising:

means for performing a significance propagation pass on a group of coefficients;

means for creating a data structure that indicates locations of coefficients in the group of coefficients that are to be processed in subsequent passes; and

means for performing a cleanup sub-bitplane pass by

accessing the data structure to obtain information to identify coefficients to be skipped for cleanup sub-bitplane pass processing,

accessing a memory storing the group of coefficient using the information to only access coefficients identified as being in the cleanup pass, and

coding cleanup bits accessed from the memory.

46. The apparatus defined in Claim 45 wherein a plurality of cleanup bits are accessed and non-cleanup bits are ignored.

47. The apparatus defined in Claim 45 wherein the group of coefficients comprises a code-block.

48. The apparatus defined in Claim 45 wherein the means for creating at least one data structure comprises means for creating a data structure to describe locations of coefficients in a cleanup sub-bitplane pass for the group of coefficients.

49. The apparatus defined in Claim 48 wherein the data structure includes an indication of each run of cleanup bits and the number of coefficients to skip in the group of coefficients before the next run of cleanup bits.

50. The apparatus defined in Claim 49 wherein at least one of the indication of each run of cleanup bits and the number of coefficients to skip is represented with a variable length code as follows:

count	codeword
1	000
2	001
3	010
4	011
5	1000000000000
6	1000000000001
...	...
4096	1111111111011

51. The apparatus defined in Claim 49 wherein at least one of the indication of each run of cleanup bits and the number of coefficients to skip is represented with a variable length code as follows:

count	codeword
1	0
2	01
3	1100
4	1101_0
5	1101_1
6	1110_0000
7	1110_0001
...	...
21	1110_1111
22	1111_0000_0000_0000
23	1111_0000_0000_0001
...	...
4096	1111_1111_1110_1010

52. The apparatus defined in Claim 49 wherein at least one of the indication of each run of cleanup bits and the number of coefficients to skip is represented with a gamma code as follows:

count	codeword (γ^1 format)	codeword (γ format)
1	0	0
2	10_0	100
3	10_1	110
4	110_00	10100
5	110_01	10110
6	110_10	11100
7	110_11	11110
8	1110_000	1010100
9	1110_001	1010110
...	...	...
15	1110_111	1111110
16	11110_0000	101010100
32	111110_00000	10101010100
64	111110_000000	1010101010100
128	11111110_0000000	101010101010100
256	111111110_00000000	10101010101010100
512	1111111110_000000000	1010101010101010100
1024	11111111110_000000000	10101010101010101010100
2048	111111111110_0000000000	101010101010101010101010100
4096	1111111111110_00000000000	10101010101010101010101010100

53. The apparatus defined in Claim 49 wherein the indication of each of run of cleanup bits is represented with a variable length code.

54. The apparatus defined in Claim 49 wherein the indication of a number of coefficients to skip is represented with a variable length code.

55. The apparatus defined in Claim 49 wherein the indication of each of run of cleanup bits and the number of coefficients to skip are represented with variable length codes.

56. An article of manufacture having one or more recordable media with executable instructions stored thereon which, when executed by a system, cause the system to:

perform a significance propagation pass on a group of coefficients;

create a data structure that indicates locations of coefficients in the group of coefficients that are to be processed in subsequent passes; and

perform a cleanup sub-bitplane pass by

accessing the data structure to obtain information to identify coefficients to be skipped for cleanup sub-bitplane pass processing,

accessing a memory storing the group of coefficient using the information to only access coefficients identified as being in the cleanup pass, and

coding cleanup bits accessed from the memory.

57. A method comprising:

a context model reading at least one data structure that indicates locations of coefficients in a group of coefficients that are to be processed during refinement or cleanup passes, the at least one data structure containing refinement bit run counts specifying the size of runs of refinement bits and refinement bit skip counts indicating numbers of coefficients to skip between the runs of refinement bits, the at least one data structure further containing cleanup bit run counts specifying the size of runs of cleanup bits and cleanup bit skip counts indicating numbers of coefficients to skip between the runs of cleanup bits; and

a context model accessing a memory based on information in the at least one data structure.

58. A decoder comprising:

a memory; and

a context model coupled to the memory, wherein the context model reads at least one data structure that indicates locations of coefficients in a group of coefficients that are to be processed during refinement or cleanup passes, the at least one data structure containing refinement bit run counts specifying the size of runs of refinement bits and refinement bit skip counts

indicating numbers of coefficients to skip between the runs of refinement bits, the at least one data structure further containing cleanup bit run counts specifying the size of runs of cleanup bits and cleanup bit skip counts indicating numbers of coefficients to skip between the runs of cleanup bits; and wherein the context model accesses a memory based on information in the at least one data structure.

59. A method comprising:

accessing a data structure to obtain information to identify coefficients to be skipped for refinement sub-bitplane pass processing, wherein the data structure indicates location of coefficients in the group of coefficients that are to be processed in subsequent passes;

accessing a memory storing the group of coefficient using the information to only access coefficients identified as being in the refinement pass; and

coding refinement bits accessed from the memory.

60. The method defined in Claim 59 further comprising performing a significance propagation pass, including coding coefficients in the significance propagation pass, while creating the data structure.

61. An apparatus comprising:

means for accessing a data structure to obtain information to identify coefficients to be skipped for refinement sub-bitplane pass processing, wherein the data structure indicates location of coefficients in the group of coefficients that are to be processed in subsequent passes;

means for accessing a memory storing the group of coefficient using the information to only access coefficients identified as being in the refinement pass; and

means for coding refinement bits accessed from the memory.

62. The apparatus defined in Claim 61 further comprising means for performing a significance propagation pass, including means for coding coefficients in the significance propagation pass, while creating the data structure.

63. A method comprising:

accessing a data structure to obtain information to identify coefficients to be skipped for refinement sub-bitplane pass processing, wherein the data structure indicates locations of coefficients in the group of coefficients that are to be processed in subsequent passes;

accessing a memory storing the group of coefficient using the information to only access coefficients identified as being in the cleanup pass; and

coding cleanup bits accessed from the memory.

64. The method defined in Claim 63 further comprising performing a significance propagation pass, including coding coefficients in the significance propagation pass, while creating the data structure.

65. An apparatus for decoding information comprising:

a memory to store run counts and skip counts of two distinct portions of the memory separated by a third portion of memory;

decoding hardware coupled to the memory to decode a run count and a skip count obtained from the memory simultaneously.

66. The apparatus defined in Claim 65 wherein the third portion of the memory comprises an unused portion of memory between the two portions of memory.

67. The apparatus defined in Claim 66 wherein the unused portion of memory is adjacent to the two portions of memory.

68. An apparatus for decoding information comprising:

a memory to store run counts and skip counts of two distinct portions of the memory separated by a third portion of memory;

decoding hardware coupled to the memory to decode a run count and a skip count obtained from the memory simultaneously.

69. The apparatus defined in Claim 68 wherein the third portion of the memory comprises an unused portion of memory between the two portions of memory.

70. The apparatus defined in Claim 69 wherein the unused portion of memory is adjacent to the two portions of memory.

71. A method for performing a significance propagation pass, the method comprising:

determining whether information being processed is in a significance propagation pass, refinement pass, or cleanup pass in response to significance state information for a region;

asserting signals indicative of whether 4x4 subregions in the region have coefficients in the significance propagation pass, refinement pass or cleanup pass;

coding bits of coefficients in the significance pass while identifying run and skip counts for refinement and cleanup passes; and

processing bits of coefficients identified as being in the refinement and cleanup passes.

72. The method defined in Claim 71 further comprising starting a new run for either the refinement or cleanup passes when a previous bit is a significance propagation bit and a current state is not significance propagation.

73. The method defined in Claim 72 further comprising adjusting an index into a table in a data structure storing run and skip count information to a new entry and initializing the new entry to a state indicative of the start of the new run.

74. The method defined in Claim 73 wherein initializing the new entry comprises setting a skip indication to zero and a run indicator to a value.

75. The method defined in Claim 71 wherein each of the subregions comprises a 4x4 region.

76. The method defined in Claim 71 further comprising counting a number of significant coefficients in each of the subregions.

77. The method defined in Claim 71 further comprising processing all bits in each of the subregions as refinement if a count equals the size of said each subregion.

78. The method defined in Claim 71 further comprising processing all bits in each of the subregions as cleanup if a count equals zero and no neighbors are significant.

79. An apparatus for performing a significance propagation pass, the apparatus comprising:

means for determining whether information being processed is in a significance propagation pass, refinement pass, or cleanup pass in response to significance state information for a region;

means for asserting signals indicative of whether subregions in the region have coefficients in the significance propagation pass, refinement pass or cleanup pass;

means for coding bits of coefficients in the significance pass while identifying run and skip counts for refinement and cleanup passes; and

means for processing bits of coefficients identified as being in the refinement and cleanup passes.

80. The apparatus defined in Claim 79 further comprising means for starting a new run for either the refinement or cleanup passes when a

previous bit is a significance propagation bit and a current state is not significance propagation.

81. The apparatus defined in Claim 80 further comprising means for adjusting an index into a table in a data structure storing run and skip count information to a new entry and means for initializing the new entry to a state indicative of the start of the new run.

82. The apparatus defined in Claim 81 wherein initializing the new entry comprises means for setting a skip indication to zero and a run indicator to a value.

83. The apparatus defined in Claim 79 wherein each of the subregions comprises a 4x4 region.

84. The apparatus defined in Claim 79 further comprising counting a number of significant coefficients in each of the subregions.

85. The apparatus defined in Claim 79 further comprising processing all bits in each of the subregions as refinement if a count equals the size of said each subregion.

86. The apparatus defined in Claim 79 further comprising processing all bits in each of the subregions as cleanup if a count equals zero and no neighbors are significant.

87. An apparatus for performing a significance propagation pass comprising:

a determine pass unit having an input coupled to receive significance state information for a first region of a first predetermined size and having a first set of output indications indicative of whether a pass associated with each coefficient in a second region of a second predetermined size is in a significance propagation pass, refinement pass, or cleanup pass;

a processing unit having inputs coupled to the output indications and having a next non-refinement bit output, a next non-cleanup bit output, and a current pass indication output; and

a control unit coupled to outputs from the processing unit, the control unit, in response thereto, to generate an indication of a next index into a refinement bit data structure, a refinement run indication, a refinement skip indication, and a significant propagation indication.

88. The apparatus defined in Claim 87 wherein the processing unit comprises:

a selection unit coupled to receive the first set of output indications and a count indicative of a current coefficient of the second region being processed, the selection logic having a second set of output indications for the current coefficient, with only one of the second set of output indications being asserted for the current coefficient;

a mask unit coupled to receive refinement and cleanup pass indications from the determine pass unit and the count indication, the mask unit having a pair of outputs representing masked versions of the refinement and cleanup pass indications by masking bits of the refinement and cleanup indications associated with coefficients in the second region that have already been processed; and

a priority encoder having inputs coupled to the masked versions of the refinement and cleanup indications, the priority encoder having a pair of outputs representing a next non-refinement bit position and a next non-cleanup bit position for the significance propagation pass.

89. The apparatus defined in Claim 87 wherein the control unit, unit, in response to outputs from the processing unit, generates an indication of a next index into a cleanup bit data structure, a cleanup run indication, and a cleanup skip indication.

90. The apparatus defined in Claim 87 wherein a coefficient is processed in the significance propagation pass if the current bit is in the significant propagation pass.

91. The apparatus defined in Claim 87 wherein K bits are processed in the refinement pass if the current bit is in the refinement pass, where K equals a run length of refinement bits.

92. The apparatus defined in Claim 89 wherein K bits are processed in the cleanup pass if the current bit is in the cleanup pass, where K equals a run length of cleanup bits.

93. The apparatus defined in Claim 87 wherein, when the next non-refinement bit is to be processed later in time than the current coefficient, the control logic sets the refinement run indication equal to the run length and sets the refinement skip indication to indicate that no amount of skipping of coefficients in the second region is to be made based on the current coefficient.

94. The apparatus defined in Claim 93 wherein the control logic sets the refinement next index indication to indicate an increment is to be made to an index into a data structure storing run and skip counts depicting refinement bit locations in a code-block.

95. The apparatus defined in Claim 89 wherein, when the next non-refinement bit is to be processed later in time than the current coefficient, the control logic sets the refinement run indication equal to the

run length and sets the refinement skip indication to indicate that no amount of skipping of coefficients in the second region is to be made based on the current coefficient;

and further wherein the control logic sets the refinement next index indication to indicate an increment is to be made to an index into a data structure storing run and skip counts depicting refinement bit locations in a code-block; and

and further wherein the control logic further sets the cleanup run indication to indicate that no current run of cleanup bits is occurring and the cleanup skip indication equal to the next non-refinement bit position minus a number equal to a position of the current coefficient in the second region.

96. The apparatus defined in Claim 95 wherein the control logic sets the cleanup next index indication to indicate that no change is to be made to an index into a data structure storing run and skip counts depicting cleanup bit locations in a code-block.

97. The apparatus defined in Claim 96 wherein the significance propagation index is set to indicate the current coefficient is not in a significance propagation pass.

98. The apparatus defined in Claim 87 wherein a coefficient is processed in the significance propagation pass if the current bit is in the significant propagation pass.

99. The apparatus defined in Claim 89 wherein, when the next non-cleanup bit is to be processed later in time than the current coefficient, the control logic sets the cleanup run indication equal to the run size and sets the cleanup skip indication to indicate that no amount of skipping of coefficients in the second region is to be made based on the current coefficient.

100. The apparatus defined in Claim 99 wherein, the control logic sets the cleanup next index indication to indicate an increment is to be made to an index into a data structure storing run and skip counts depicting cleanup bit locations in a codeblock.

101. The apparatus defined in Claim 100 further sets the cleanup run indication to indicate that no current run of cleanup bits is occurring and the cleanup skip indication equal to the next non-refinement bit position minus a number equal to a position of the current coefficient in the second region.

102. The apparatus defined in Claim 101 wherein the control logic sets the cleanup next index indication to indicate that no change is to be made to an index into a data structure storing run and skip counts depicting cleanup bit locations in a code-block.

103. The apparatus defined in Claim 102 wherein the significance propagation index is set to indicate the current coefficient is not in a significance propagation pass.

104. The apparatus defined in Claim 87 wherein the determine pass unit examines a predetermined number of NxM regions, one for each coefficient being processed, where N and M are integers.

105. The apparatus defined in Claim 104 wherein the NxM regions comprise 3x3 regions.

106. The apparatus defined in Claim 104 wherein the predetermined number of NxM regions comprises sixteen.

107. The apparatus defined in Claim 89 wherein the first set of output indications comprises the significance propogation pass, refinement pass, and cleanup pass output indications and each of the significance propogation pass, refinement pass, and cleanup pass output indications is 16 bits wide.

108. The apparatus defined in Claim 89 wherein each of significance propagation pass, refinement pass, and cleanup pass output indications has one bit corresponding to a coefficient in the second region, and only one of such bit is asserted during each cycle.

109. The apparatus defined in Claim 89 wherein the masked unit masks signal lines of each of refinement pass and cleanup pass indications corresponding to coefficients that have already been processed.

110. The apparatus defined in Claim 88 wherein the priority encoder comprises a zero-finding priority encoder.